UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/716,916	11/18/2003	Greg Christopher JR.	07844-602001-P555	6410

21876 7590 12/07/2006

FISH & RICHARDSON P.C.
P.O. Box 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
CHEN, QING	

ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 12/07/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/716,916	CHRISTOPHER, GREG	
	Examiner	Art Unit	
	Qing Chen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 September 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office action is in response to the amendment filed on September 26, 2006.

2. **Claims 1-24** are pending.

3. **Claims 2, 9, 18, and 22** have been amended.

4. The objections to the specification are withdrawn in view of Applicant's amendments to the specification.

5. The objections to Claims 2, 9, and 22 are withdrawn in view of Applicant's amendments to the claims. However, the objection to Claim 18 is maintained and further explained below.

6. The 35 U.S.C. § 112, first paragraph, rejection of Claim 1 is withdrawn in view of Applicant's persuasive arguments.

7. The 35 U.S.C. § 112, second paragraph, rejection of Claim 22 is withdrawn in view of Applicant's amendments to the specification.

8. The 35 U.S.C. § 101 rejection of Claims 1-3 and 24 are withdrawn in view of Applicant's persuasive arguments. However, the 35 U.S.C. § 101 rejections of Claims 10-12 are maintained and further explained below.

Response to Amendment

Claim Objections

9. **Claim 18** is objected to because of the following informalities:

- **Claim 18** contains a typographical error: the last comma (,) in the series should be added between the words "additions" and "and." Applicant is advised to make the

correction in order to keep the grammatical style consistent throughout the specification and claims.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

10. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

11. **Claims 10-12** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The result of **Claims 10 and 12** is directed to the act of "identifying," which does not appear to be a tangible result so as to constitute a practical application of the idea. The act of "identifying" is merely a thought or an abstract idea and does not appear to produce a tangible result even if the step of identification does occur, since the result of that identification is not conveyed in the real world. The result is an identification, which is neither used in a disclosed practical application nor made available for use in a disclosed practical application. It also does not appear that the usefulness of the identification can be realized from the claimed steps to support a disclosed specific, substantial, and credible utility so as to produce a useful result.

In addition, the result of **Claim 11** is directed to the act of "designating," which also does not appear to be a tangible result so as to constitute a practical application of the idea. The reasoning is stated above in the aforementioned paragraph and applied in the same manner.

Therefore, the claims do not meet the statutory requirement of 35 U.S.C. § 101, since the claims are not directed to a practical application of the § 101 judicial exception producing a result tied to the physical world.

Claim Rejections - 35 USC § 102

12. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

13. **Claims 1-6, 10, 12-15, and 24** are rejected under 35 U.S.C. 102(e) as being anticipated by **Breggin et al. (US 6,560,776)**.

As per **Claim 1**, Breggin et al. disclose:

- creating data that represents a new expectation for an installation result, for one or more resources associated with a software installer, the new expectation being a transition from an expectation of volatility to an expectation of stability for future software installs *(see Figure 4; Column 3: 60-63, "The embodiment is an installation analysis tool that creates a verification or installation database based on an analysis of the install program before the installation."; Column 8: 24-50, "The first location 300a corresponds to a discrepancy (between the installation database and the actual location on the target computer): the second position 300b*

to a discrepancy in file size; the third position 300c to a discrepancy in file modification date; the fourth position 300d to a discrepancy in file version (applies where the target computer has an older file version); the fifth position 300e to a discrepancy in file version (applies where the target computer has a newer file version); and finally the sixth position 300f to a discrepancy in registry information.").

As per **Claim 2,** the rejection of **Claim 1** is incorporated; and Breggin et al. further disclose:

- generating a comparison of a current software installation with a previous software installation *(see Figure 3A: 228; Column 9: 29-37, "... the computer compares the installation database to the installed database or file to identify differences or discrepancies.")*; and

- identifying, based on the comparison, resources that have changed in their installation result from the previous software installation to the current software installation, despite the new expectation of stability for the resources *(see Figure 5; Column 9: 55-66, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format. An "exception" is typically a difference between corresponding fields in the installation and installed databases or files."; Column 10: 3-14, "Where an upper and lower entry are in the same column for a given file, it means that an exception exists in that area." and "For example, entry 500a refers to the modification date for the file in the installation database and entry 500b refers to a different modification date for the file on the target computer.").*

As per **Claim 3**, the rejection of **Claim 2** is incorporated; and Breggin et al. further disclose:

- identifying, based on the comparison, resources that have not changed in their installation result from the previous software installation to the current software installation, despite an expectation that the unchanged resources should change from the previous software installation to the current software installation *(see Figure 5; Column 9: 55-66, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format. An 'exception' is typically a difference between corresponding fields in the installation and installed databases or files."; Column 10: 9-15, "Where a column has no lower entry for a given file, there is no exception for the file in that area." and "For the file 504 there is no discrepancy for file location, file size, file version and registration status.").*

As per **Claim 4**, the rejection of **Claim 2** is incorporated; and Breggin et al. further disclose:

- presenting potential problems with the current software installation based on the identified resources to facilitate verification of an installer for a software product *(see Figure 5; Column 10: 17-28, "... exceptions can be displayed selectively at differing levels depending, for example, on the field to which the exception pertains.").*

As per **Claim 5**, the rejection of **Claim 4** is incorporated; and Breggin et al. further disclose:

- tracking expectations for the resources in a primary installation baseline and a secondary installation baseline, and wherein presenting the potential problems comprises presenting a baseline-update interface by transmitting markup language data *(see Column 10: 40-42, "In Web-based applications, the installed database or file can be incorporated into one or more web pages." and 49-67 through Column 11: 1-5, "In this process, a baseline file, which is simply a 'snapshot' of the exceptions on the target computer at a given time, is generated manually or automatically. The baseline file can be used to 'mask' or remove previous exceptions from the installed file or database." and "This feature permits a user to track which files have changed and how they have changed in a manner that permits subsequent (or cascading) changes to be installed." and "... after the user has selected the baselining option ... the processor in box 240 opens and reads the baseline file(s). In box 244, the processor iteratively compares the contents of the baseline file(s) with the list of exceptions and other pertinent information in the installed database of file(s).").*

As per **Claim 6**, the rejection of **Claim 4** is incorporated; and Breggin et al. further disclose:

- excluding a set of resources from the generated comparison for the software product *(see Column 3: 14-15, "The exceptions can be filtered to exclude known exceptions from analysis."; Column 11: 5-8, "Any matching items are removed from the list of exceptions to be displayed graphically to the user.").*

As per **Claim 10**, Breggin et al. disclose:

- generating a comparison of a current software installation with a previous software installation *(see Figure 3A: 228; Column 9: 29-37, "... the computer compares the installation database to the installed database or file to identify differences or discrepancies.")*; and

- identifying based on the comparison, resources that have not changed in their installation result from the previous software installation to the current software installation, despite an expectation that the unchanged resources should change from the previous software installation to the current software installation *(see Figure 5; Column 9: 55-66, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format. An 'exception' is typically a difference between corresponding fields in the installation and installed databases or files."; Column 10: 9-15, "Where a column has no lower entry for a given file, there is no exception for the file in that area." and "For the file 504 there is no discrepancy for file location, file size, file version and registration status.")*.

As per **Claim 12**, the rejection of **Claim 10** is incorporated; and Breggin et al. further disclose:

- identifying, based on the comparison, resources that have changed in their installation result from the previous software installation to the current software installation, despite an expectation that the changed resources should not change from the previous software installation to the current software installation *(see Figure 5; Column 9: 55-66, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format. An*

"exception" is typically a difference between corresponding fields in the installation and installed databases or files."; Column 10: 3-14, "Where an upper and lower entry are in the same column for a given file, it means that an exception exists in that area." and "For example, entry 500a refers to the modification date for the file in the installation database and entry 500b refers to a different modification date for the file on the target computer.").

As per **Claim 13**, the rejection of **Claim 12** is incorporated; and Breggin et al. further disclose:

- presenting potential problems with the current software installation based on the identified resources to facilitate verification of an installer for a software product *(see Figure 3B: 236; Figure 5; Column 9: 55-58, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format."; Column 10: 17-28, "As illustrated in by FIG. 5, exceptions can be displayed selectively at differing levels depending, for example, on the field to which the exception pertains.").*

As per **Claim 14**, the rejection of **Claim 13** is incorporated; and Breggin et al. further disclose:

- tracking the expectations of resource changes in a primary installation baseline and a secondary installation baseline, and wherein presenting the potential problems comprises presenting a baseline-update interface by transmitting markup language data *(see Column 10: 40-42, "In Web-based applications, the installed database or file can be incorporated into one or*

more web pages." and 49-67 through Column 11: 1-5, "In this process, a baseline file, which is simply a 'snapshot' of the exceptions on the target computer at a given time, is generated manually or automatically. The baseline file can be used to 'mask' or remove previous exceptions from the installed file or database." and "This feature permits a user to track which files have changed and how they have changed in a manner that permits subsequent (or cascading) changes to be installed." and "... after the user has selected the baselining option ... the processor in box 240 opens and reads the baseline file(s). In box 244, the processor iteratively compares the contents of the baseline file(s) with the list of exceptions and other pertinent information in the installed database of file(s).").

As per **Claim 15**, the rejection of **Claim 13** is incorporated; and Breggin et al. further disclose:

- excluding a set of resources from the generated comparison for the software product *(see Column 3: 14-15, "The exceptions can be filtered to exclude known exceptions from analysis."; Column 11: 5-8, "Any matching items are removed from the list of exceptions to be displayed graphically to the user.").*

As per **Claim 24**, Breggin et al. disclose:

- means for generating a current install comparison of a computing system before a software installation, with the computing system after the software installation, the current install comparison identifying new resources that are added to the computing system during the software installation and identifying system resources that are modified during the software

installation, and the current install comparison recording at least one attribute of the resources for the current software installation *(see Figure 3A: 228; Figure 5; Column 9: 29-44, "... the computer compares the installation database to the installed database or file to identify differences or discrepancies. The computer then proceeds to box 230 where the computer reads registry information on the target computer and, in box 321, compares this information with registry information in the installation database to identify other differences and discrepancies." and "... the processor collects target computer information. This information includes a summary of the installation process (i.e., installation log which is a directory of all files that were replaced and/or all files added during the installation), hardware configuration information, operating system configuration, current date/time, and the like.")*;

- means for generating a software trend comparison of the current install comparison with a previous install comparison, the software trend comparison including which of the resources have changed in the at least one attribute from the previous install to the current install, and the software trend comparison indicating which of the resources have not changed in the at least one attribute from the previous install to the current install *(see Figure 5; Column 9: 55-66, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format. An 'exception' is typically a difference between corresponding fields in the installation and installed databases or files."; Column 10: 3-15, "Where an upper and lower entry are in the same column for a given file, it means that an exception exists in that area." and "For example, entry 500a refers to the modification date for the file in the installation database and entry 500b refers to a different modification date for the file on the target computer." and*

"Where a column has no lower entry for a given file, there is no exception for the file in that area." and "For the file 504 there is no discrepancy for file location, file size, file version and registration status.");

- means for comparing the software trend comparison with a record of installation expectations that indicates which of the resources should be in flux, and which of the resources should be stable from the previous install to the current install, with respect to the at least one attribute *(see Figure 5; Column 9: 55-66, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format. An 'exception' is typically a difference between corresponding fields in the installation and installed databases or files."; Column 10: 3-15, "Where an upper and lower entry are in the same column for a given file, it means that an exception exists in that area." and "For example, entry 500a refers to the modification date for the file in the installation database and entry 500b refers to a different modification date for the file on the target computer." and "Where a column has no lower entry for a given file, there is no exception for the file in that area." and "For the file 504 there is no discrepancy for file location, file size, file version and registration status.")*; and

- means for presenting potential problems with the current software installation based on the comparison of the software trend comparison with the expectations record *(see Figure 3B: 236; Figure 5; Column 9: 55-58, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format."; Column 10: 17-28, "As illustrated in by FIG. 5,*

exceptions can be displayed selectively at differing levels depending, for example, on the field to which the exception pertains.").

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. **Claims 7-9 and 16-18** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Breggin et al. (US 6,560,776).**

As per **Claim 7**, the rejection of **Claim 4** is incorporated; and Breggin et al. further disclose:

- wherein the expectations of resource changes, including the data, are stored in a relational database indexed by date *(see Column 8: 24-29, "The database lists ... last modification date of the file ('DATE') ...").*

However, Breggin et al. do not disclose:

- wherein the expectations of resource changes, including the data, are stored in a relational database indexed by platform, language, and product configuration.

Official Notice is taken that it is old and well known within the computing art to index data in a relational database using various attributes. Data in a database is often indexed by

various attributes pertaining to the particular application of the data. For example, software installation data is often indexed in a database by platform (operating system), supported languages, and product configuration information. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the expectations of resource changes, including the data, are stored in a relational database indexed by platform, language, and product configuration. The modification would be obvious because one of ordinary skill in the art would be motivated to store and access additional useful data in the database pertaining to the software installation.

As per **Claim 8**, the rejection of **Claim 4** is incorporated; and Breggin et al. further disclose:

- wherein the expectations for the resources relate to attributes comprising modification date stamp information and file size information *(see Column 8: 24-29, "The database lists ... file size ('SIZE') ... last modification date of the file ('DATE') ...").*

However, Breggin et al. do not disclose:

- wherein the expectations for the resources relate to attributes comprising security permissions information and checksum information.

Official Notice is taken that it is old and well known within the computing art to define data in a database using various attributes. Data in a database often contains various attributes pertaining to the particular application of the data. For example, software installation data in a database often contains file permission information and file checksum information. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to

include wherein the expectations for the resources relate to attributes comprising security permissions information and checksum information. The modification would be obvious because one of ordinary skill in the art would be motivated to provide additional useful data pertaining to the software installation to the user.

As per **Claim 9**, the rejection of **Claim 4** is incorporated; and Breggin et al. further disclose:

- wherein the resources comprise files and system registry entries, and the installation result comprises modifications of the resources *(see Column 7: 23-25, "... the processor ... creates a list of program files , data files, and registry entry changes."; Column 10: 61-64, "... files have changed and how they have changed in a manner that permits subsequent (or cascading) changes to be identified.")*.

However, Breggin et al. do not disclose:

- the installation result comprises deletions and additions of the resources.

Official Notice is taken that it is old and well known within the computing art to indicate the additions and deletions of resources in an installation result. It is common for a software installation process to output added and deleted files to indicate changes in the installed program. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the installation result comprises deletions and additions of the resources. The modification would be obvious because one of ordinary skill in the art would be motivated to provide additional installation information that can be used for system monitoring.

As per **Claim 16**, the rejection of **Claim 13** is incorporated; and Breggin et al. further disclose:

- wherein the expectations of resource changes are stored in a relational database indexed by date *(see Column 8: 24-29, "The database lists ... last modification date of the file ('DATE') ...")*.

However, Breggin et al. do not disclose:

- wherein the expectations of resource changes are stored in a relational database indexed by platform, language, and product configuration.

Official Notice is taken that it is old and well known within the computing art to index data in a relational database using various attributes. Data in a database is often indexed by various attributes pertaining to the particular application of the data. For example, software installation data is often indexed in a database by platform (operating system), supported languages, and product configuration information. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the expectations of resource changes are stored in a relational database indexed by platform, language, and product configuration. The modification would be obvious because one of ordinary skill in the art would be motivated to store and access additional useful data in the database pertaining to the software installation.

As per **Claim 17**, the rejection of **Claim 13** is incorporated; and Breggin et al. further disclose:

- wherein the expectations of resource changes relate to attributes comprising modification date stamp information and file size information *(see Column 8: 24-29, "The database lists ... file size ('SIZE') ... last modification date of the file ('DATE') ... ").*

However, Breggin et al. do not disclose:

- wherein the expectations of resource changes relate to attributes comprising security permissions information and checksum information.

Official Notice is taken that it is old and well known within the computing art to define data in a database using various attributes. Data in a database often contains various attributes pertaining to the particular application of the data. For example, software installation data in a database often contains file permission information and file checksum information. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the expectations of resource changes relate to attributes comprising security permissions information and checksum information. The modification would be obvious because one of ordinary skill in the art would be motivated to provide additional useful data pertaining to the software installation to the user.

As per **Claim 18**, the rejection of **Claim 13** is incorporated; and Breggin et al. further disclose:

- wherein the resources comprise files and system registry entries, and the installation result comprises modifications of the resources *(see Column 7: 23-25, "... the processor ... creates a list of program files , data files, and registry entry changes."; Column 10: 61-64, "...*

files have changed and how they have changed in a manner that permits subsequent (or cascading) changes to be identified.").

However, Breggin et al. do not disclose:

- the installation result comprises deletions and additions of the resources.

Official Notice is taken that it is old and well known within the computing art to indicate the additions and deletions of resources in an installation result. It is common for a software installation process to output added and deleted files to indicate changes in the installed program. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the installation result comprises deletions and additions of the resources. The modification would be obvious because one of ordinary skill in the art would be motivated to provide additional installation information that can be used for system monitoring.

16. **Claims 11, 19, 20, 22, and 23** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Breggin et al. (US 6,560,776)** in view of **Kruger et al. (US 6,738,970)**.

As per **Claim 11**, the rejection of **Claim 10** is incorporated; and Breggin et al. further disclose:

- designating a new expectation of stability for the specified resources according to the received input *(see Figure 4; Column 3: 60-63, "The embodiment is an installation analysis tool that creates a verification or installation database based on an analysis of the install program before the installation."; Column 8: 24-50, "The first location 300a corresponds to a discrepancy (between the installation database and the actual location on the target computer):*

the second position 300b to a discrepancy in file size; the third position 300c to a discrepancy in file modification date; the fourth position 300d to a discrepancy in file version (applies where the target computer has an older file version); the fifth position 300e to a discrepancy in file version (applies where the target computer has a newer file version); and finally the sixth position 300f to a discrepancy in registry information.").

However, Breggin et al. do not disclose:

- receiving input specifying which of the identified resources should be static in their installation result for future software installations.

Kruger et al. disclose:

- receiving input specifying which of the identified resources should be static in their installation result for future software installations *(see Column 12: 37-40, "Same node pruner 256 removes from the supertree all nodes marked as 'same' because the objects corresponding to these nodes will not need to be altered during subsequent installations.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Kruger et al. into the teaching of Breggin et al. to include receiving input specifying which of the identified resources should be static in their installation result for future software installations. The modification would be obvious because one of ordinary skill in the art would be motivated to prevent errors from occurring during subsequent software installations.

As per **Claim 19**, Breggin et al. disclose:

- a build controller *(see Column 4: 1-6, "... the build computer.")*; and

- an install controller comprising a database including a baseline recording expectations of stability or volatility for one or more resources associated with a software installer *(see Figure 1: 200; Column 7: 47-49, "... the processor places the information into the installation database or file."; Column 10: 50-52, "The baseline file can be used to 'mask' or remove previous exceptions from the installed file or database.")*;

- wherein the build controller automatically triggers the install controller to initiate installer tests as part of a software build process, and collects test results to be presented in a report comprising a baseline-update interface *(see Figure 3B: 236; Figure 5; Column 4: 16-21, "... the (build) computer first reads in ... the installation program or script ... and creates a list of program files, data files, and/or registry entry changes ... and writes certain of this information to the installation database."; Column 9: 55-58, "In box 236, all of the information obtained in the comparing steps 228 and 231, including exceptions and collected information about the target computer, is graphically displayed in any desirable format."; Column 10: 17-28, "As illustrated in by FIG. 5, exceptions can be displayed selectively at differing levels depending, for example, on the field to which the exception pertains.")*.

However, Breggin et al. do not disclose:

- one or more install slave machines, and the install controller automatically dispatches installation to the one or more install slave machines.

Kruger et al. disclose:

- one or more install slave machines, and the install controller automatically dispatches installation to the one or more install slave machines *(see Column 4: 1-5, "The master computer is any computer on which the computer software can be properly installed, and for which such*

installation will be used as a model for installation of the software on other computer systems." and *19-27, "The systems (sic) sends the instructions, files, and program to other computer systems using conventional management software ... ").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Kruger et al. into the teaching of Breggin et al. to include one or more install slave machines, and the install controller automatically dispatches installation to the one or more install slave machines. The modification would be obvious because one of ordinary skill in the art would be motivated to provide redundant data backup or testing platforms for diagnosing and monitoring software installation/performance.

As per **Claim 20**, the rejection of **Claim 19** is incorporated; however, Breggin et al. do not disclose:

- wherein the one or more slave machines comprise multiple computers.

Kruger et al. disclose:

- wherein the one or more slave machines comprise multiple computers *(see Column 4: 1-5, "The master computer is any computer on which the computer software can be properly installed, and for which such installation will be used as a model for installation of the software on other computer systems.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Kruger et al. into the teaching of Breggin et al. to include wherein the one or more slave machines comprise multiple computers. The

modification would be obvious because one of ordinary skill in the art would be motivated to properly install and test software programs.

As per **Claim 22**, the rejection of **Claim 19** is incorporated; and Breggin et al. further disclose:

- wherein the baseline-update interface comprises a web-based user interface allowing baseline updates across SKU, language, operating system, and custom/non-custom installs, in combination or all at once *(see Column 10: 40-42, "In Web-based application, the installed database or file can be incorporated into one or more web pages.")*.

As per **Claim 23**, the rejection of **Claim 19** is incorporated; and Breggin et al. further disclose:

- wherein the expectations of resource changes relate to attributes comprising modification date stamp information and file size information *(see Column 8: 24-29, "The database lists ... file size ('SIZE') ... last modification date of the file ('DATE') ...")*.

However, Breggin et al. do not disclose:

- wherein the expectations of resource changes relate to attributes comprising security permissions information and checksum information.

Official Notice is taken that it is old and well known within the computing art to define data in a database using various attributes. Data in a database often contains various attributes pertaining to the particular application of the data. For example, software installation data in a database often contains file permission information and file checksum information. Therefore, it

would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the expectations of resource changes relate to attributes comprising security permissions information and checksum information. The modification would be obvious because one of ordinary skill in the art would be motivated to provide additional useful data pertaining to the software installation to the user.

17. **Claim 21** is rejected under 35 U.S.C. 103(a) as being unpatentable over **Breggin et al. (US 6,560,776)** in view of **Kruger et al. (US 6,738,970)** as applied to Claim 19 above, and further in view of **Suorsa et al. (US 2002/0156831)**.

As per **Claim 21**, the rejection of **Claim 19** is incorporated; however, Breggin et al. and Kruger et al. do not disclose:

- wherein the install controller communicates with the one or more install slave machines using Simple Object Access Protocol.

Suorsa et al. disclose:

- wherein the install controller communicates with the one or more install slave machines using Simple Object Access Protocol *(see Paragraph [0052], "... messages that are exchanged between the gateway and the agents can be in the form of remote procedure calls that conform to the XML-RPC protocol, or the Simple Object Access Protocol (SOAP).").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Suorsa et al. into the teaching of Breggin et al. to include wherein the install controller communicates with the one or more install slave

machines using Simple Object Access Protocol. The modification would be obvious because one of ordinary skill in the art would be motivated to provide a way to communicate between applications running on different operating systems with different technologies and programming languages.

Response to Arguments

18. Applicant's arguments filed on September 26, 2006 have been fully considered but they are not persuasive.

In the remarks, Applicant argues that:

a) Claim 10, as a whole, produces a "useful, concrete and tangible result" for use by a user by "generating a comparison" and "identifying" "resources that have not changed" for tracking and verifying software installation.

Examiner's response:

a) "... the focus is not on whether the steps taken to achieve a particular result are useful, tangible, and concrete, but rather on whether the final result achieved by the claimed invention is "useful, tangible, and concrete." See MPEP § 2106 and the 35 U.S.C. 101 rejections of Claims 10-12 above.

In the remarks, Applicant argues that:

b) Breggin simply does not teach the limitations of "new expectation for an installation result" and "the new expectation being a transition from an expectation of volatility to an expectation of stability for future software installs" as recited in claim 1.

Examiner's response:

b) Breggin et al. clearly disclose the limitations of "new expectation for an installation result" and "the new expectation being a transition from an expectation of volatility to an expectation of stability for future software installs" as recited in Claim 1 *(see Figure 4; Column 3: 60-63, "The embodiment is an installation analysis tool that creates a verification or installation database based on an analysis of the install program before the installation."; Column 8: 24-50, "The first location 300a corresponds to a discrepancy (between the installation database and the actual location on the target computer): the second position 300b to a discrepancy in file size; the third position 300c to a discrepancy in file modification date; the fourth position 300d to a discrepancy in file version (applies where the target computer has an older file version); the fifth position 300e to a discrepancy in file version (applies where the target computer has a newer file version); and finally the sixth position 300f to a discrepancy in registry information.").*

In the remarks, Applicant argues that:

c) Breggin's identification of "exception" stands in sharp contract with "identifying ... resources that have not changed" as recited in claim 10.

Examiner's response:

c) Examiner disagrees. Breggin et al. clearly disclose "identifying ... resources that have not changed" as recited in Claim 10 *(Column 10: 9-15, "Where a column has no lower entry for a given file, there is no exception for the file in that area." and "For the file 504 there is no discrepancy for file location, file size, file version and registration status.")*. Note that no exception for the file would indicate that the file has not changed, since an exception would define any discrepancy.

In the remarks, Applicant argues that:

d) Breggin merely teaches that "a baseline file, which is simply a "snapshot" of the exceptions on the target computer at a given time, is generated manually or automatically." (10:50-52.) Breggin simply does not teach "tracking expectations for the resources in a primary installation baseline and a secondary installation baseline ..." as recited in claim 5.

Examiner's response:

d) Breggin et al. clearly disclose "tracking expectations for the resources in a primary installation baseline and a secondary installation baseline ..." as recited in Claim 5 *(see Column 10: 61-64, "This feature permits a user to track which files have changed and how they have changed in a manner that permits subsequent (or cascading) changes to be installed."; Column 10: 66-67 through Column 11: 1-5, "... after the user has selected the baselining option ... the processor in box 240 opens and reads the baseline file(s). In box 244, the processor iteratively*

compares the contents of the baseline file(s) with the list of exceptions and other pertinent information in the installed database of file(s).").

In the remarks, Applicant argues that:

e) Further, claim 24 recites, among other features: "the software trend comparison indicating which of the resources have not changed in the at least one attribute from the previous install to the current install; ... which of the resources should be in flux, and which of the resources should be stable from the previous install to the current install." Thus, claim 24 is patentable for at least similar reasons to those discussed above in connection with claims 1, 5, and 10.

Examiner's response:

e) Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

In the remarks, Applicant argues that:

f) Furthermore, the Office uses improper hindsight reconstruction to reject claims 7-9 and 16-18 because the only motivation for the missing elements of Breggin is Applicant's own disclosure and claim language.

Examiner's response:

f) In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

In the remarks, Applicant argues that:

g) Regarding independent claim 19, the Office mistakenly equates Breggin's teaching of an "installation database" (Figure 1, element 200; 7:47-49) with the element of "install controller" recited in claim 19. The Applicant discloses that "[a]n install controller 420 manages the install testing process using the techniques described above and a database 430. The installer controller 420 can (sic) and/or the system under test 410 can communicate results to the database 430." (page 8, paragraph [0030].) Therefore, the claimed "install controller" is not the same as an "installation database."

Examiner's response:

g) Examiner disagrees. The claimed "install controller" is the same as an "installation database," since the claim recites that the install controller comprises a database. Furthermore, although the claims are interpreted in light of the specification, limitations from the specification

are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

In the remarks, Applicant argues that:

h) Further, the Office mistakenly equates Breggin's teaching of "the (build) computer ... creates a list of program files ... and writes certain of this information to the installation database" (4:16-21) with the element of "the build controller automatically triggers the install controller to initiate installer tests ..." recited in claim 19. The Applicant discloses that "The build controller 510 communicates with the install controller 520 to trigger installer tests, and in response, the install controller 520 automatically obtains installers from the build controller 510." (page 11, paragraph [0038].) Contrarily, Breggin's teaching of "creat[ing]" and "writ[ing]" to the "installation database" is not the same as "trigger[ing] the install controller to initiate installer tests" as in claim 19. Therefore, the suggested Breggin-Kruger combination does not teach or suggest all the elements of claim 19, and claim 19 should be allowed.

Examiner's response:

h) Examiner disagrees. Breggin et al.'s teaching of "creat[ing]" and "writ[ing]" to the "installation database" is the same as "trigger[ing] the install controller to initiate installer tests," since creating and writing a list of program files, data files, and/or registry entry changes to the installation database to initiate the installation process clearly indicates that the install controller is "triggered" to initiate installer tests. Furthermore, although the claims are interpreted in light

of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Conclusion

19. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

A. **Muller, Nathan J.**, "Focus on OpenView: A Guide to Hewlett Packard's Network and Systems Management Platform," 1995, CBM Books, pg. 179-185 and 210-211.

20. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The

Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

WEI ZHEN
SUPERVISORY PATENT EXAMINER

QC / qc
November 30, 2006